

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-163742

(43)Date of publication of application : 18.06.1999

(51)Int.Cl.

H03M 13/12

(21)Application number : 09-324380

(71)Applicant : NEC CORP

(22)Date of filing : 26.11.1997

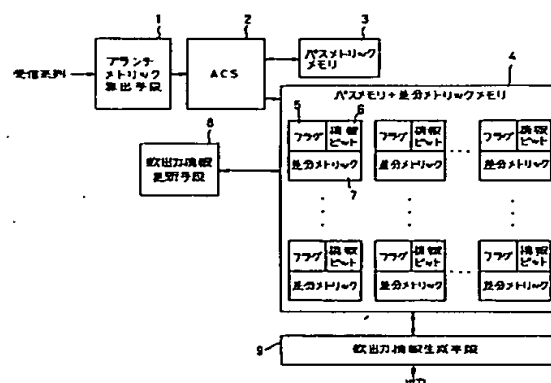
(72)Inventor : OKAMURA TOSHIHIKO

## (54) SOFT OUTPUT DECODER

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To attain high speed processing at high accuracy for soft output decoding of a convolution code.

**SOLUTION:** A flag area 5, an information bit area 6, a difference metric area 7 are provided in a path metric + difference metric memory 4, and sets of the areas 5-7 are assigned corresponding to nodes. Each flag area 5 corresponds to respective node on a trellis line drawing of a convolution code and stores an index that denotes whether it is a node on a survival maximum likelihood path passing through the corresponding node or it has been used for updating a soft output. Whether or not the update processing has been finished is discriminated, based on the result of a difference metric comparison between the index and a current update value. Furthermore, a soft output is corrected efficiently by preparing a cross-reference table for mean values of square sums of soft outputs, communication line constants and normalization constants.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-163742

(43)公開日 平成11年(1999) 6月18日

(51)Int.Cl.<sup>6</sup>

H 0 3 M 13/12

識別記号

F I

H 0 3 M 13/12

審査請求 有 請求項の数 7 O L (全 10 頁)

(21)出願番号 特願平9-324380

(22)出願日 平成 9 年(1997)11月26日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

(72)発明者 岡村 利彦

東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

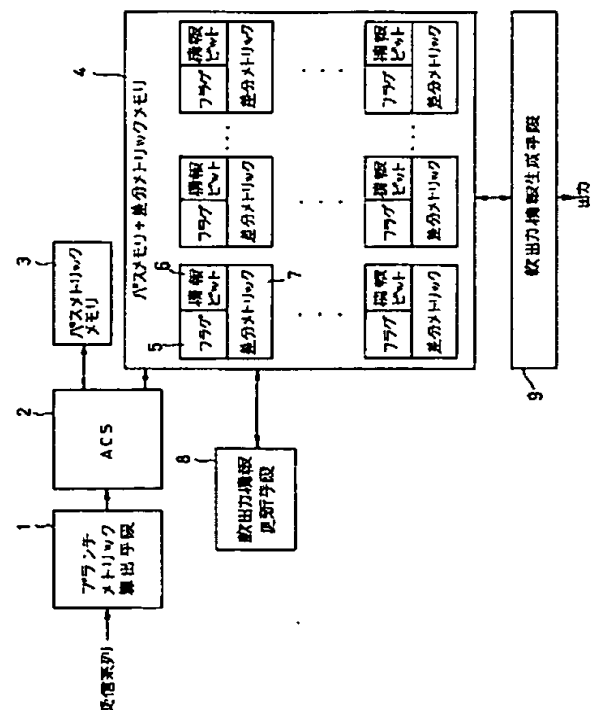
(74)代理人 弁理士 ▲柳▼川 信

(54)【発明の名称】 軟出力復号装置

(57)【要約】

【課題】 畳込み符号の軟出力復号の高速化及び高精度化を図る。

【解決手段】 パスメトリック+差分メトリックメモリ 4 に、フラグ領域 5、情報ビット領域 6、差分メトリック領域 7 を設け、これ等各領域 5～7 の組をノード対応に割当てておく。フラグ領域 5 は畳込み符号のトレリス線図上のノードの対応し、その対応ノードを通る生き残り最尤パス上のノードであるか、または軟出力更新に利用したかどうかを示すインデックスを保持する。このインデックスと現在の更新値との差分メトリックの比較結果により、更新処理終了判断を行うことができる。また、軟出力の平方和の平均値、通信路定数、更には正規化定数の対応表を準備することで、効率良く軟出力の補正処理を行うことができる。



## 【特許請求の範囲】

【請求項1】 畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、前記畳込み符号のトレリス線図上の各ノードに対してフラグ領域、情報ビット領域、差分メトリック領域が夫々一組として割当てられ、前記フラグ領域には、対応ノードがヴィタビ復号により最終的に選択されたパス上にあるかまたは当該ノードの時点における軟出力更新処理において試されたことがあるかを表す情報を保持し、前記情報ビット領域には、対応ノードでヴィタビ復号により選択される前記トレリス線図上の一時点前の状態を表すインデックスを保持し、前記差分メトリック領域には、前記トレリス線図上において対応ノードでヴィタビ復号により選択されたパスと放棄されたパスとのメトリックの差を保持したメモリ手段と、前記フラグ領域、情報ビット領域、差分メトリック領域に保持されている情報に基づきヴィタビ復号において最終的に選択されたパス上の対応するノードの情報ビットの軟出力を生成する軟出力生成手段と、を含むことを特徴とする軟出力復号装置。

【請求項2】 前記差分メトリック領域を前記軟出力によって順次書き換えるようにしたことを特徴とする請求項1記載の軟出力復号装置。

【請求項3】 前記ヴィタビ復号において選択されたパスを表す情報ビット列と対応する軟出力情報を保持する軟出力メモリを更に含み、前記軟出力メモリを前記軟出力の更新に従って順次書き換えるようにしたことを特徴とする請求項1記載の軟出力復号装置。

【請求項4】 前記畳込み符号はフィードバック型の畳込み符号であり、前記情報ビット領域は前記トレリス線図上の対応するノードにおける前記ヴィタビ復号で選択されたパスの情報ビットを保持するようにしたことを特徴とする請求項1～3いずれか記載の軟出力復号装置。

【請求項5】 前記軟出力生成手段は、前記トレリス線図上で前記ヴィタビ復号により最終的に選択されたパス上の各ノードで放棄されたパスである対抗パスのメトリックと共に、この対抗パス上のノードで前記ヴィタビ復号で放棄されたパスのメトリックを利用して情報ビットの軟出力を生成するようにしたことを特徴とする請求項3記載の軟出力復号装置。

【請求項6】 畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、軟出力の平方和の平均値と通信路の雑音を定める通信路定数及び軟出力値の正規化定数との対応表と、実際の受信系列の復号処理にて生じた軟出力の平方和の平均値から前記対応表を用いて前記通信路定数及び前記正規化定数を求める手段と、実際の軟出力を前記通信路定数及び前記正規化定数を使

用して補正する手段、とを含むことを特徴とする軟出力復号装置。

【請求項7】 畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、前記畳込み符号のトレリス線図上の各ノードに対してフラグ領域、情報ビット領域、差分メトリック領域が夫々一組として割当てられ、

前記フラグ領域には、対応ノードがヴィタビ復号により最終的に選択されたパス上にあるかまたは当該ノードの時点における軟出力更新処理において試されたことがあるかを表す情報を保持し、

前記情報ビット領域には、対応ノードでヴィタビ復号により選択される前記トレリス線図上の一時点前の状態を表すインデックスを保持し、

前記差分メトリック領域には、前記トレリス線図上において対応ノードでヴィタビ復号により選択されたパスと放棄されたパスとのメトリックの差を保持したメモリ手段と、

前記フラグ領域、情報ビット領域、差分メトリック領域に保持されている情報に基づきヴィタビ復号において最終的に選択されたパス上の対応するノードの情報ビットの軟出力を生成する軟出力生成手段と、

前記軟出力の平方和の平均値と通信路の雑音を定める通信路定数及び軟出力値の正規化定数との対応表と、実際の受信系列の復号処理にて生じた軟出力の平方和の平均値から前記対応表を用いて前記通信路定数及び前記正規化定数を求める手段と、

実際の軟出力を前記通信路定数及び前記正規化定数を使用して補正する手段、とを含むことを特徴とする軟出力復号装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は軟出力復号装置に関し、特に入力アルファベットが2値である畳込み符号のヴィタビ復号処理を使用した軟出力復号装置に関するものである。

## 【0002】

【従来の技術】 畳込み符号は誤り訂正符号化方式として現在広く利用されている。詳しくは「符号理論、今井秀樹著、1990年1月、電子情報通信学会」の第11、12章に記述されている。ここでは簡単な例を用いて説明する。

【0003】 図11は畳込み符号の符号器の構成を示したものである。特にこの符号器はレート1/2のフィードバック型と呼ばれる畳込み符号化器の最も簡単な例である。100、101はメモリ（記憶素子）である。時点 $t$ におけるその値を $s(t, 0)$ 、 $s(t, 1)$ とする。

【0004】 時点 $t$ の入力ビット $u(t)$ に対して2ビットの出力 $w(t, 0)$ 、 $w(t, 1)$ が生成される。

## 3

図11は組織符号となっており、 $w(t, 0)$ は、  
 $w(t, 0) = u(t)$

$$\begin{aligned} w(t, 1) &= s(t, 0) + s(t, 1) + u(t) + s(t, 1) \\ &= s(t, 0) + u(t) \end{aligned}$$

となる。ここで、“+”は排他論理和を表す。

【0006】時点 $(t+1)$ のメモリの内容は、

$$s(t+1, 1) = s(t, 0)$$

$$s(t+1, 0) = s(t, 0) + s(t, 1) + u(t)$$

となる。

【0007】この様に符号器の内部に $s(t, 0)$ 、 $s(t, 1)$ という記憶素子を持ち、この値(“状態”と呼ぶ)に依存して出力ビット $w(t, 0)$ 、 $w(t, 1)$ が決まるところが畳込み符号の特徴である。記憶素子の数に1を加算した値は拘束長と呼ばれ、畳込み符号を特徴付ける値の一つである。

【0008】畳込み符号の出力と次の状態への遷移は、現在の符号器の状態と入力ビットに依存する。それを時系列的に表現した図はトレリス線図と呼ばれる。図12は図11の畳込み符号に対するトレリス線図を表している。図12において、各段は上から順に $(s(t, 0), s(t, 1))$ が $(0, 0)$ 、 $(1, 0)$ 、 $(0, 1)$ 、 $(1, 1)$ となっている状態を表している。また各列は左から時点0, 1, 2, …の状態を表している。トレリス線図上の各時点における各状態をノードと呼ぶ。

【0009】時点0は初期状態を表す。通常 $s(0, 0) = s(0, 1)$ を初期状態とする。時点1では時点0から遷移可能な状態のみをとりうる。 $u(1) = 0$ の場合は $(0, 0)$ から $(0, 0)$ へ状態遷移が起こる。またこの時 $w(1, 0) = w(1, 1) = 0$ となる。また、 $u(1) = 1$ の場合は $(0, 0)$ から $(1, 0)$ へ状態遷移が起こり、 $w(1, 0) = w(1, 1) = 1$ となる。トレリス線図は状態遷移が可能なきはノード間をブランチ(線分)で結んだ図である。図12で各ブランチに振ってあるインデックスは順にその状態遷移に付随する出力ビット $w(t, 0)$ 、 $w(t, 1)$ と情報ビット $u(t)$ を表している。

【0010】トレリス線図上で連続するブランチを選択したものをパスと呼ぶ。畳込み符号の符号語となるビット列はパス上の出力ビット列に制限されている。各時点に対応する受信系列とブランチの間の尤度に応じたメトリックが計算される。これをブランチメトリックと呼ぶ。パスに対してもメトリックとしてブランチメトリックの和が供せられる。これをパスメトリックと呼ぶ。ここではメトリックが大きい程尤度が高い様にメトリックを設定する。

【0011】畳込み符号の復号は通信路上などでエラーが混入した受信系列から実際に送信された符号語を推定する処理である。符号語 $W$ が等確率で送信されるとき、

## 4

となる。

【0005】一方、 $w(t, 1)$ は、

受信系列 $Y$ に対して $W$ の下での $Y$ の条件付き確率 $P(Y|W)$ を最大化する符号語を実際に送信された符号語と推定する復号法を最尤復号法と呼ぶ。この確率は通信路に応じて決まる。これを実行するアルゴリズムはヴィタビ復号法として知られている。ヴィタビ復号はトレリス線図上の各ノードについて、そこへ到達するパスメトリック最大のパスを選択し(生き残りパスと呼ぶ)、もう一方のパスを放棄することにより時点を追って順に効率的に最尤復号を行うことができる。開始当初などを除いて各時点において、状態数のパスが存在することになる。

【0012】近年、符号語の推定だけでなく、推定された符号語に対応する各情報ビットの信頼度情報に相当する軟出力が必要となる場合が生じている。特に重要な応用はターボ符号の名前で知られる、フィードバック型畳込み符号の並列連接符号の復号への適用である。これに関しては「1996年10月、アイ・イー・イー・トラランザクション・オン・コミュニケーション、第44巻、第10号、1261～1271頁(IEEE Transactions on Communication, Vol.44, No.10, pp.1261-1271)」に述べられている。

【0013】受信系列を $Y$ 、 $k$ 番目の情報ビットを $u(k)$ とする。理想的な信頼度情報(軟出力) $L(u(k))$ は次で与えられる値である。

$$L(u(k)) = \log P(u(k) = 0 | Y) / P(u(k) = 1 | Y)$$

$L(u(k))$ を求めるアルゴリズムはMAP(Maximum A Posterior probability)アルゴリズムと呼ばれ、前述の文献に詳述されている。但し、MAPは復号遅延、装置規模が大きい等の問題がある。

【0015】 $L(u(k))$ の近似値を求めることで上記の問題を解決するための手段として、ヴィタビ復号法を利用したSOVA(Soft Output Viterbi Alogrithm)が知られている。これに関する一つの装置化方法が「1993年5月、プロシーディング・オブ・インターナショナル・コンファレンス・オブ・コミュニケーション、737～740頁、Proceeding of International Conference of Communication, pp.737-740」に記述されている。図8はそれを模式的に表したものである。

【0016】図8は軟出力装置を表している、ブランチメトリック算出手段80、ACS(Add Compare Select)81、パスメトリックメモリ82は通常のヴィタビ復号と同様の構成である。大きな違いはパスメモリの部分で、図8ではパスメモリは差分メトリックメモリ85と併設されている。通常のパスメモリは85の情報ビット領域のみからなる。この差分メトリック領域には生き残り

りバスと放棄されたバスのメトリックの差をトレリス線図の各ノードに対して持つ。軟出力情報は軟出力情報メモリ87に保持される。

【0017】軟出力生成方法を図9に従って述べる。図9は、時点 $t$ までの最尤バスがヴィタビ復号により確定したとき、 $x := d(t, s)$ を用いた軟出力情報更新方法を示している。ここで、 $d(t, s)$ は時点 $t$ 、状態 $s$ のノードにおける生き残りバスと放棄されるバスのメトリックの差である。

【0018】まず初期設定を行う(ステップ90)。ここで $s$ は最尤バスの時点 $t$ での状態である。このノードで放棄されたヴィタビ復号における生き残りバスを $V$ とする。 $V$ の時点 $(t-1)$ の状態を $u$ とする。最尤バスの時点 $t$ での情報ビットを $b(t)$ とおく。 $x$ を用いて軟出力情報更新処理を行う時点の下限を $t_0$ とおく。その幅 $T_1$ は予め決めておく。

【0019】必要に応じて時点 $t$ の軟出力情報を更新する(ステップ91)。 $L(t)$ は時点 $t$ の軟出力情報を表すとする。フィードバック型の場合は $L(t)$ に $x$ を、そうでない場合は $L(t)$ を適切な値で初期化する。

【0020】時点をつつ戻し $(t-1)$ とし、 $V$ における $v$ の一つ前のノードを $v'$ とする(ステップ92)。 $v'$ から $v$ への遷移に対応する情報ビットと時点 $t$ における最尤バスの情報ビット $b(t)$ が等しいかどうかと比較する(ステップ92)。ステップ92の条件が否定された場合は $L(t)$ と $x$ を比較する(ステップ94)。

【0021】もし、 $x$ の方が $L(t)$ より小さかったら $L(t)$ を $x$ で置換える(ステップ95)。時点 $t$ が $t_0$ に達したら、更新処理を終了する(ステップ96)。更新処理を継続する時には $v'$ を $v$ としてステップ92に戻る(ステップ97)。

【0022】これをトレリス線図上で図示すると図10の様になる。図9において $s(t_3)$ 、 $s(t_4)$ 、 $s(t-1)$ 、 $s(t)$ 、 $s(t_1)$ 、 $s(t_2)$ で示されるバスが最尤バスだとする。時点 $t$ での軟出力情報を決定するのは時点 $t_1$ 、 $t_2$ といった $t$ から先で最尤バスから放棄されたバスである。 $s(t_1)$ で放棄されたバス $s(t_1)$ 、 $v_2$ 、 $v_1$ 、 $v_0$ 、 $s(t_4)$ で示されるバスは時点 $t$ での情報ビットが異なるため、それ以前の $L(t)$ と $d(t, s(t_1))$ を比較して $d(t, s(t_1))$ の方が小さかったら前者を後者で置換える。一方、 $s(t_2)$ 、 $v_6$ 、 $v_5$ 、 $v_4$ 、 $s(t_3)$ で示される、 $s(t_2)$ で放棄されたバスは時点 $t$ での情報ビットが最尤バスに等しいため、 $d(t_2, s(t_2))$ は $L(t)$ に影響を与えない。

【0023】

【発明が解決しようとする課題】上述のSOVAではトレリス線図全体を保持する必要がなく、その一部を保持すれば十分なのでMAPと比較して装置規模を大きく取

ることができる。ヴィタビ復号を利用した軟出力情報生成処理では、最尤バスが確定した後の軟出力情報更新のために対抗バスをトレースバックする処理が通常のヴィタビ復号と比較して最大の負荷になる。このトレースバックの回数を小さくすることが課題となる。

【0024】また、MAPと違い、軟出力情報生成のために使用するバスが制限されている(最尤バスの対抗バスのみ)ため、その軟出力情報の精度を高めることが課題となる。使用するバスの範囲を計算量を増やさずに拡大することが一つの方法であり、前述の文献に触れられている。

【0025】もう一つの方法は最終的に得られた軟出力情報にある種の正規化を行うことが一つの方法であり、「1996年5月、プロシーディング・オブ・インターナショナル・コンファレンス・オブ・コミュニケーション、102~106頁、Proceeding of International Conference of Communication, pp.102-106」に記述されている。しかし、この正規化は通信路における雑音の影響に依存する処理であり、実際の復号処理においてこの通信路の雑音の大きさを推定しなければならないことが一つの課題となる。

【0026】本発明の目的は、畳込み符号の軟出力復号処理の高速化及び高精度化を可能とした軟出力復号装置を提供することである。

【0027】

【課題を解決するための手段】本発明によれば、入力アルファベットが2値である畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、前記畳込み符号のトレリス線図上の各ノードに対してフラグ領域、情報ビット領域、差分メトリック領域が夫々組として割当てられ、前記フラグ領域には、対応ノードがヴィタビ復号により最終的に選択されたバス上にあるかまたは当該ノードの時点における軟出力更新処理において賦されたことがあるかを表す情報を保持し、前記情報ビット領域には、対応ノードでヴィタビ復号により選択される前記トレリス線図上の一時点前の状態を表すインデックスを保持し、前記差分メトリック領域には、前記トレリス線図上において対応ノードでヴィタビ復号により選択されたバスと放棄されたバスとのメトリックの差を保持したメモリ手段と、前記フラグ領域、情報ビット領域、差分メトリック領域に保持されている情報に基づきヴィタビ復号において最終的に選択されたバス上の対応するノードの情報ビットの軟出力を生成する軟出力生成手段とを含むことを特徴とする軟出力復号装置が得られる。

【0028】そして、前記差分メトリック領域を前記軟出力によって順次替換えるようにしたことを特徴とし、また、前記ヴィタビ復号において選択されたバスを表す情報ビット列と対応する軟出力情報を保持する軟出力メモリを更に含み、前記軟出力メモリを前記軟出力の更新に従って順次替換えるようにしたことを特徴としてい

る。

【0029】更に、前記畳込み符号はフィードバック型の畳込み符号であり、前記情報ビット領域は前記トレリス線図上の対応するノードにおける前記ヴィタビ復号で選択されたパスの情報ビットを保持するようにしたとを特徴とし、また、前記軟出力生成手段は、前記トレリス線図上で前記ヴィタビ復号により最終的に選択されたパス上の各ノードで放棄されたパスである対抗パスのメトリックと共に、この対抗パス上のノードで前記ヴィタビ復号で放棄されたパスのメトリックを利用して情報ビットの軟出力を生成するようにしたことを特徴としている。

【0030】更に本発明によれば、入力アルファベットが2値である畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、軟出力の平方和の平均値と通信路の雑音を定める通信路定数及び軟出力値の正規化定数との対応表と、実際の受信系列の復号処理にて生じた軟出力の平方和の平均値から前記対応表を用いて前記通信路定数及び前記正規化定数を求める手段と、実際の軟出力を前記通信路定数及び前記正規化定数を使用して補正する手段、とを含むことを特徴とする軟出力復号装置が得られる。

【0031】更にはまた、本発明によれば、入力アルファベットが2値である畳込み符号のヴィタビ復号処理を使用した軟出力復号装置であって、前記畳込み符号のトレリス線図上の各ノードに対してフラグ領域、情報ビット領域、差分メトリック領域が夫々一組として割当てられ、前記フラグ領域には、対応ノードがヴィタビ復号により最終的に選択されたパス上にあるかまたは当該ノードの時点における軟出力更新処理において試されたことがあるかを表す情報を保持し、前記情報ビット領域には、対応ノードでヴィタビ復号により選択される前記トレリス線図上の一時点前の状態を表すインデックスを保持し、前記差分メトリック領域には、前記トレリス線図上において対応ノードでヴィタビ復号により選択されたパスと放棄されたパスとのメトリックの差を保持したメモリ手段と、前記フラグ領域、情報ビット領域、差分メトリック領域に保持されている情報に基づきヴィタビ復号において最終的に選択されたパス上の対応するノードの情報ビットの軟出力を生成する軟出力生成手段と、前記軟出力の平方和の平均値と通信路の雑音を定める通信路定数及び軟出力値の正規化定数との対応表と、実際の受信系列の復号処理にて生じた軟出力の平方和の平均値から前記対応表を用いて前記通信路定数及び前記正規化定数を求める手段と、実際の軟出力を前記通信路定数及び前記正規化定数を使用して補正する手段とを含むことを特徴とする軟出力復号装置が得られる。

【0032】

【発明の実施の形態】以下に本発明の実施例について図面を参照しつつ詳述する。

【0033】図1は本発明の実施例の構成を示しており、ランチメトリック算出手段1、ACS (Add-Compare-Select) 2、バスメトリックメモリ3、バスメモリ+差分メトリックメモリ4、軟出力情報更新手段8、軟出力情報生成手段9からなる。バスメモリ+差分メトリックメモリ4の部分はフラグ領域5、情報ビット領域6、差分メトリック領域7に分けることができる。

【0034】ランチメトリック算出手段、ACS、バスメトリックメモリは通常のヴィタビ復号と同様に、トレリス線図上の次の時点の各ノード(状態)へ到達する最適なパスを選択、不要な場合はパスを放棄する処理を行う。

【0035】時点 $t$ におけるパス選択を行った場合、ACS 2からバスメモリ+差分メトリックメモリ4へ送られる情報は時点 $t$ におけるトレリス線図における各ノードに対して次の2点である。第一にそのノードへ延ばすことが可能な二本のパスのうち、最適な方を示すインデックスである。そして第二にその二本のパスのメトリックの差である差分メトリックである。ノード $v$ におけるこの差分メトリックを $d(t, v)$ と表すことにする。

【0036】バスメモリ+差分メトリックメモリ4はトレリス線図の一部(ある限られた範囲の時点分のトレリス)、または全体のノードに対応するフラグ領域5、情報ビット領域6、差分メトリック領域7からなる。バスメモリ+差分メトリックメモリ4はACS 2から送られた情報を夫々対応するノードに割当てられた情報ビット領域6、差分メトリック領域7に書き込む。

【0037】この際、フラグ領域5は初期化をしておく。ここでは、“0”を書き込むことにする。時点 $t$ におけるACS 2の処理が終了した時には、バスメモリ+差分メトリックメモリ4の時点 $t$ のノードに対応する領域が全て上記の様に更新される。

【0038】バスメモリ+差分メトリックメモリ4は、最尤パス決定のための遅延幅 $T_0$ と軟出力情報決定のための遅延幅 $T_1$ の和である( $T_0 + T_1$ )時点分のトレリスの各ノードに対するフラグ領域5、情報ビット領域6、差分メトリック領域7を最低でも保持する必要がある。

【0039】本発明の特徴は差分メトリックと併せてフラグ領域を持つことが特徴であり、これと差分メトリック領域の書き換えを行うことにより可能となる軟出力判定更新処理の効率化について図2に従って述べる。

【0040】時点 $(t + T_0)$ でのACS 2の出力によるバスメモリ+差分メトリックメモリ4の更新が終了したら、時点 $t$ における最尤パスのノードを決定することができる(ステップ20)。これは通常のヴィタビ復号と同様である。この様にして決定された時点 $t$ のノードを $s$ 、情報ビットを $b(t)$ とする。

【0041】また時点 $t$ 、ノード $s$ に対応するフラグ領域 $f(t, s)$ を“2”とする。また差分メトリック $d$

10

20

30

40

50

( $t, s$ ) を  $x$ 、時点  $t$ 、ノード  $s$  においてヴィタビ復号において放棄されたパス（ここでは最尤パスに対する“対抗パス”と呼ぶ）を  $V$ 、 $V$  の時点 ( $t-1$ ) におけるノードを  $v$  とする。 $V$  を逆に辿りながら、この  $x$  を用いて  $t-1, t-2, \dots, t_0 := t-T+1$  における軟出力を更新する。

【0042】まず、必要に応じて時点  $t$  の軟出力更新処理を行う（ステップ21）。フィードバック型の畳込み符号の場合はこの処理が行われる。時点を一つ戻し、 $V$  上の  $v$  の一つ前のノードを  $v'$  とする（ステップ22）。ここで、 $V$  上の時点  $t$  のノードが  $v$ 、時点 ( $t-1$ ) のノードが  $v'$  となっている。

【0043】フラグ  $f(t, v)$  が2であったらこのパスはヴィタビ復号により選択された最尤パスに合流したことを意味するため、 $x$  による軟出力の更新は終了する（ステップ23）。フラグ  $f(t, v)$  が1であったらこのノードは軟出力更新処理において過去に訪れたノードであることを示す（ステップ24）。

【0044】 $f(t, v)$  が1であったら  $x$  と  $d(t, v)$  を比較する（ステップ25）。もし、 $x$  が  $d(t, v)$  より小でなかったら  $v$  から先のパスは現在の  $V$  以下の小さなメトリックで訪れたことがあることを示す。よって、ここで  $x$  を用いた軟出力更新処理を終了する。

【0045】ステップ24において  $f(t, v)$  が0であったらこのノードは軟出力情報更新処理で初めて訪れたノードであることを示す。 $f(t, v)$  を1に書換える（ステップ26）。ステップ25において  $x$  が  $d(t, v)$  より小、もしくはステップ26を経過したときには時点  $t$  における軟出力処理を更新する（ステップ27）。

【0046】 $t$  が  $t_0$  以下となったら  $x$  による軟出力更新処理は終了する（ステップ28）。更新処理を継続する時には  $v'$  を  $v$  としてステップ22に戻る。この様にフラグ領域を持ち、差分メトリック領域の書換えを行うことにより、ステップ22, 23, 24の判断で無駄な軟出力更新処理を行わずに済むのである。

【0047】ステップ27において、軟出力更新処理は  $d(t, v)$  を  $x$  で置換える処理のみでも行うことができる。このときは時点  $t$  の軟出力を出力するとき、フラグ  $f(t, v)$  が1で、情報ビットが最尤パスの情報ビットと異なるノードの中で最小の  $d(t, v)$  を出力することにより行われる。フィードバック型の畳込み符号の場合はステップ21の処理が行われるためフラグ  $f(t, v)$  が2であるノードも考慮に入れる。

【0048】従来方法と同様に、図3に示す如く、軟出力をパスメモリ+差分メモリ4と別の領域3に持つこともできる。このときにはステップ27で軟出力情報領域33を書換える。このとき差分メトリック領域7は書換える必要はない。情報ビット領域32は最尤パスに対応する情報ビットを保持する。

【0049】軟出力の精度を向上させるためにはより広範囲のパスを対象とすることが必要となる。従来方法では最尤パスに合流するヴィタビ復号における対抗パスのみを対象としてきた。しかし、フィードバック型の畳込み符号では処理量を増加させずに有効に対象となるパスの範囲を拡大して、軟出力の精度を向上させることができる。以下その方法を図4に従って説明する。

【0050】図4は時点  $t$  までの最尤パスが確定したときの軟出力生成方法を示している。また、図4は図3または図8に示した様に軟出力情報メモリを持つ場合を想定している。よって、 $d(t, v)$  の書換えは行わない。時点  $t$  の軟出力情報領域31に保持される値を  $L(t)$  とする。また時点  $t$  の情報ビット領域32に保持される値を  $b(t)$  とする。

【0051】第一に軟出力情報更新のための初期化処理を行う（ステップ41）。 $s$  により最尤パス（ヴィタビアルゴリズムで選択されるパス）時点  $t$  での状態を表すとする。 $b(t)$  に最尤パスにおける時点  $t$  での情報ビットを代入する。時点  $t$  における対向パスを  $V$  とおく。 $V$  の時点 ( $t-1$ ) での状態を  $v$  とおく。

【0052】フィードバック型の畳込み符号で、状態を表すメモリ数とフィードバックを表す多項式の次数が等しい時には一つの状態へ状態遷移を起こす情報ビットは必ず異なる。よって図2のステップ21にあたる処理が実行され、 $L(t)$ （時点  $t$  における軟出力情報）は  $d(t, s)$  で初期化される（ステップ42）。この  $d(t, s)$  はこのフローにおける軟出力情報更新の元となる値であり、 $x$  とおく。

【0053】時点  $t$  を  $t$  から ( $t-1$ ) に移し、対抗パス  $V$  における  $v$  の直前の状態を  $v'$  とおく（ステップ43）。 $v'$  から  $v$  への状態遷移に対応する情報ビットが  $b(t)$ （最尤パスにおけるこの時点での情報ビット）に等しいかどうかを判断する（ステップ44）。ステップ44で条件が成り立たない場合には、 $x$  を軟出力の更新のための候補と考え、 $y$  とおく（ステップ45）。

【0054】ステップ44で条件が成り立つ場合には、 $x$  に時点  $t$  における差分メトリック  $d(t, v)$  を加算した値 ( $x + d(t, v)$ ) を軟出力更新のための候補と考え、 $y$  とおく（ステップ46）。 $L(t)$  と  $y$  の大小を判断する（ステップ47）。 $L(t)$  よりも  $y$  が小さい場合には、 $L(t)$  を  $y$  で置換える（ステップ48）。

【0055】 $x$  による軟出力更新処理が終了かどうか判断する（ステップ49）。この終了判断には図2のフローで述べた方法が使用できる。ステップ9で軟出力更新処理を継続する場合には、 $v'$  を  $v$  としてステップ3に戻り、1時点前の軟出力更新処理を行う（ステップ10）。

【0056】この様に対象となるパスを拡大した軟出力生成方法を図で示すと図5のようになる。図5はトレリス

線図上の幾つかのパスを表しており、 $s(t3)$ ,  $s(t4)$ , ...,  $s(t2)$  で示される情報ビットが全て0のパスが最尤パスであるとする。時点 $t$ での軟出力を生成するために、従来は $s(t1)$ ,  $v2$ ,  $v1$ ,  $v0$ ,  $s(t4)$  や  $s(t2)$ ,  $v6$ ,  $v5$ ,  $v4$ ,  $v3$ ,  $s(t3)$  で示されるパスのみを扱ってきた。時点 $t$ においては、後者のパスは情報ビットが最尤パスのビットと同一のため、軟出力生成のために貢献してこなかった。

【0057】しかし、時点 $t$ で $v5$ で放棄されたパスは情報ビット1を伴っており、 $s(t2)$ ,  $v6$ ,  $v5$ ,  $v7$ ,  $v3$ ,  $s(t3)$  で表されるパスと最尤パスとのメトリックの差は $d(t2, s(t2)) + d(t, v5)$  で表される。この様なパスも扱うことにより、より精度の高い軟出力情報を生成できるのである。

【0058】また、図4、図5に示した軟出力生成方法の発展形として、更に対象となるパスの数を増やす方法を図6に示す。図6は最尤パスへの対応パスの終点ノード $v3$ ,  $v12$ などで放棄されたパスも対象とできることを示している。この時、 $s(t1)$ ,  $v12$ ,  $v17$ ,  $v16$ ,  $v15$ ,  $v13$ ,  $v9$ ,  $s(t3)$  で表されるパスと最尤パスとの差分メトリックは $d(t2, s(t12)) + d(t, v12)$  となる。

【0059】ここで、 $t(v12)$  は図5における $v12$ の時点である。また、 $s(t1)$ ,  $v3$ ,  $v2$ ,  $v1$ ,  $v5$ ,  $v4$ ,  $s(t4)$  で示されるパスと最尤パスとの差分メトリックは $d(t1, s(t1)) + d(t, v3)$ ,  $v3) + d(t, v1)$  となる。ここで $t(v3)$  は図5における $v3$ の時点である。

【0060】グィタビ復号に基づく軟出力の精度を向上させる方法の一つとして、それに正規化定数を乗じることによって補正する手段がある。この正規化定数は通信路の雑音の状態に基づいて決定される。逆に通信路の雑音がガウス分布に従う場合、軟出力の平方値の平均を利用して雑音の状態及び正規化定数を決定することができる。以下その方法について説明する。

【0061】変調方式としてBPSKを用いるとする。軟出力もガウス分布に従うと仮定する。この時1が送信された場合の軟出力の平均値を $E$ 、分散を $V$ とする。 $-1$ が送信された場合の軟出力の平均値は $-E$ 、分散は $V$ となる。実際には1,  $-1$ が混在した形で送信されるが、1,  $-1$ が送信される頻度に関わらず軟出力の平方の平均値は、 $K = V + E * E$  となる。

【0062】一般に $E$ と $V$ は正の比例に近い傾向にあり、 $E$ は通信路の雑音の大きさと1対1に対応するため、 $K = V + E * E$ も通信路の雑音の大きさと1対1に対応する。よって $K$ を計算することにより、通信路の雑音の大きさ(通信路定数)及びそれに付随する軟出力の

正規化定数を求めることができる。これ等の関係を事前にテーブル化しておくことにより実際上は効率良く処理を行うことができる。

【0063】図7は軟出力補正手段の構成図である。軟出力復号手段70、平方和平均算出手段71、正規化定数・通信路定数対応表72、補正処理手段73からなる。

【0064】軟出力復号手段70は図1等 に示した装置を使用することができる。平方和平均算出手段71は軟出力復号手段70の平方値の平均値を求める。この値に基づいて正規化定数・通信路定数対応表72の中の該当する値を選択する。通信路定数は通信路における雑音の大きさを表す量である。補正処理手段73は軟出力手段70の出力に正規化定数を乗じて補正処理を行う。並列連接化符号に使用する際には、通信路定数で決まる値を乗じる処理も行う。

【0065】

【発明の効果】本発明によれば、パスメモリ+差分メトリックメモリ内にフラグ領域を装備することにより、有効に1回の軟出力更新における終了判断ができ、対抗パスをトレースバックする回数がほぼ1/2程度になるという効果がある。

【0066】また、図3の様に軟出力生成のための対象となるパスの領域を拡大することにより、並列連接化符号の復号において従来方法と比較して0.3dB程度の符号化利得が得られる(ビットエラーレート=1/10000)。この時、処理量の増加はほとんど無視できる。

【0067】更にはまた、図7の様に軟出力の補正処理を行うことにより、並列連接化符号の復号において従来方法と比較して0.3dB程度の符号化利得が得られる(ビットエラーレート=1/10000)。また、事前に通信路の特性を知る必要がないという利点がある。

【図面の簡単な説明】

【図1】本発明の構成を示すブロック図である。

【図2】本発明による軟出力更新の終了判断の処理の流れを示すフローチャートである。

【図3】本発明の別の構成を示すブロック図である。

【図4】本発明による軟出力生成方法を示すフローチャートである。

【図5】図4の軟出力生成方法をトレリス線図上で図示している。

【図6】図5の発展型をトレリス線図上で図示している。

【図7】本発明における軟出力補正処理の構成を示すブロック図である。

【図8】従来の軟出力復号装置のブロック図である。

【図9】従来の軟出力生成方法を示すフローチャートである。

【図10】図9の軟出力生成方法をトレリス線図上で図示している。



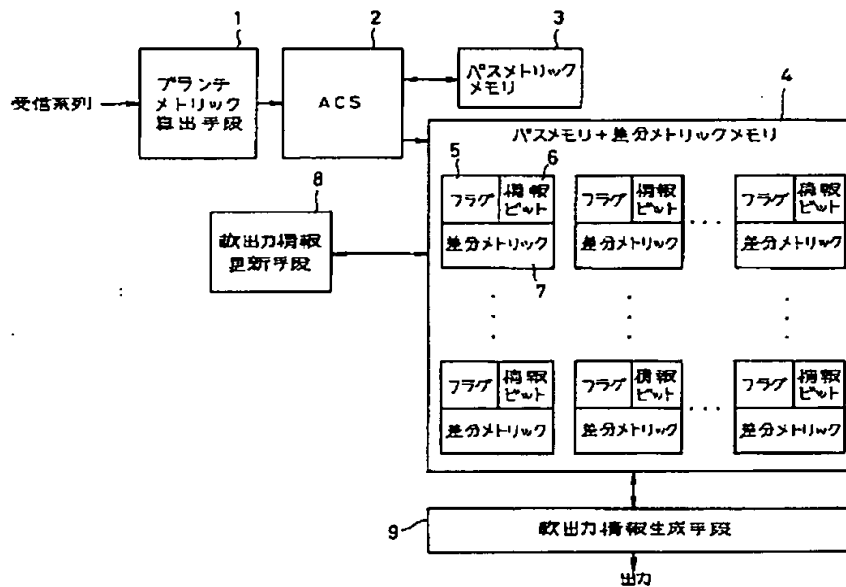
【図11】フィードバック型の畳込み符号器のブロック図である。

【図12】図11の符号に対応するトレリス線図である。

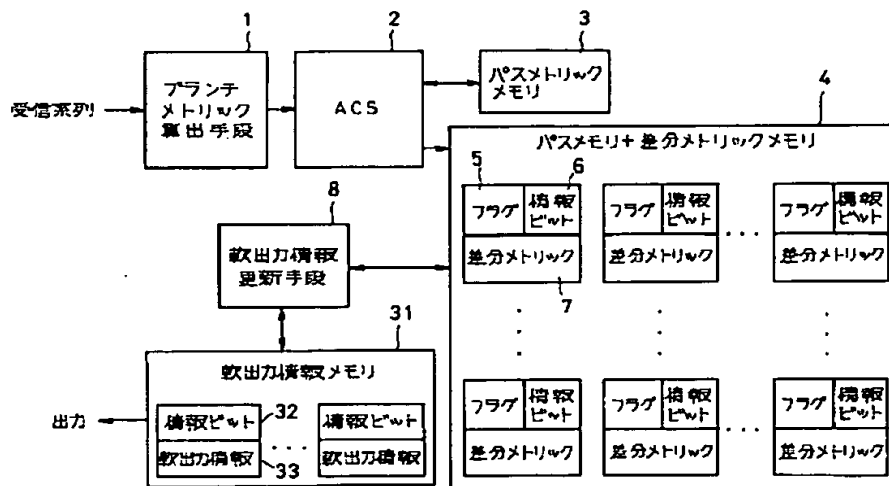
【符号の説明】

- |   |                  |       |                |
|---|------------------|-------|----------------|
| 1 | ブランチメトリック算出手段    | 6, 32 | 情報ビット領域        |
| 2 | ACS              | 7     | 差分メトリック領域      |
| 3 | パスメトリックメモリ       | 8     | 軟出力情報更新手段      |
| 4 | パスメモリ+差分メトリックメモリ | 9     | 軟出力情報生成手段      |
| 5 | フラグ領域            | 31    | 軟出力情報メモリ       |
|   |                  | 33    | 軟出力情報領域        |
|   |                  | 70    | 軟出力復号手段        |
|   |                  | 71    | 平方和平均算出手段      |
|   |                  | 72    | 正規化定数・通信路定数対応表 |
|   |                  | 10 73 | 補正処理手段         |

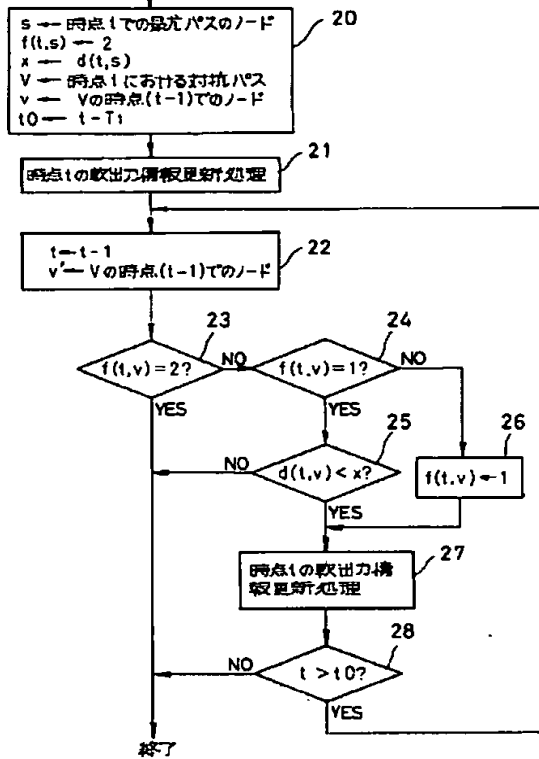
【図1】



【図3】

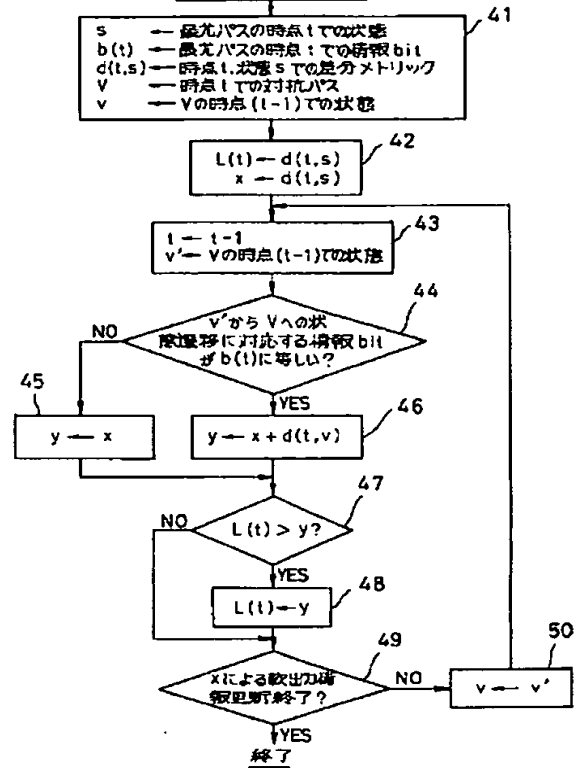


【図 2】

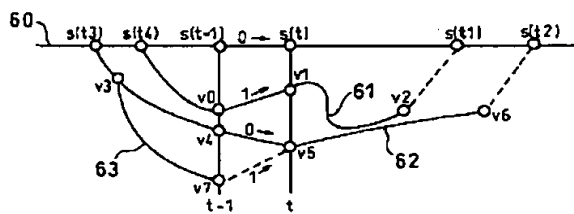
時点  $t$  での差分メトリックによる数出力情報更新開始

【図 4】

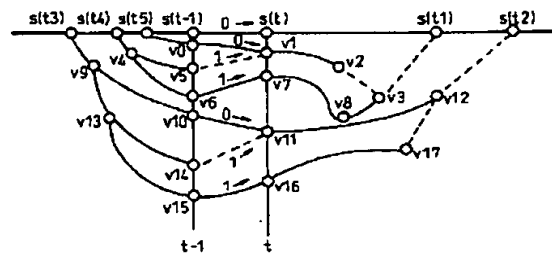
いまの最尤パス確定



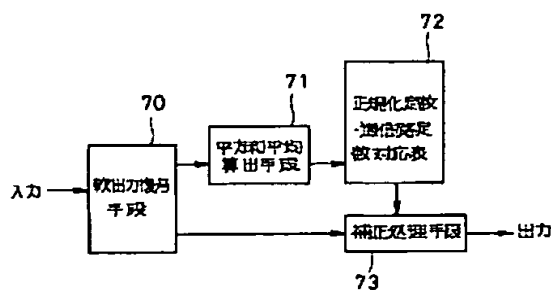
【図 5】



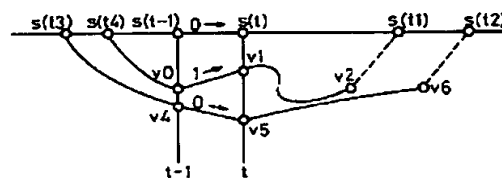
【図 6】



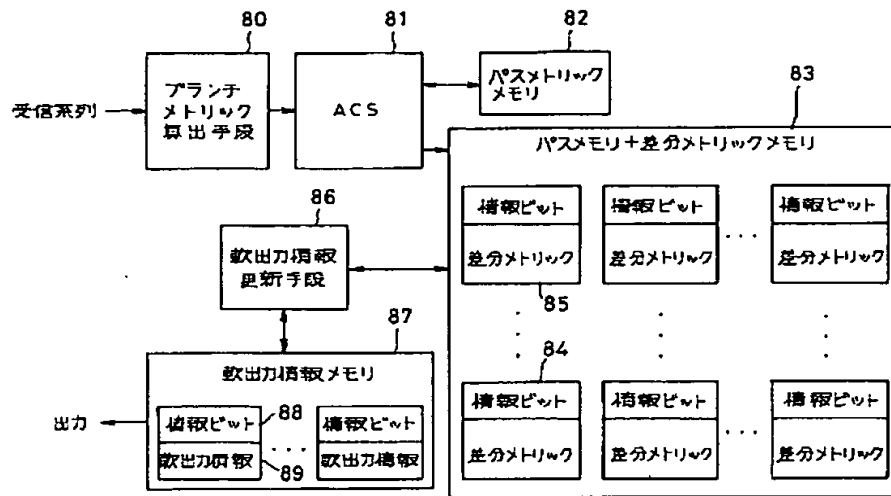
【図 7】



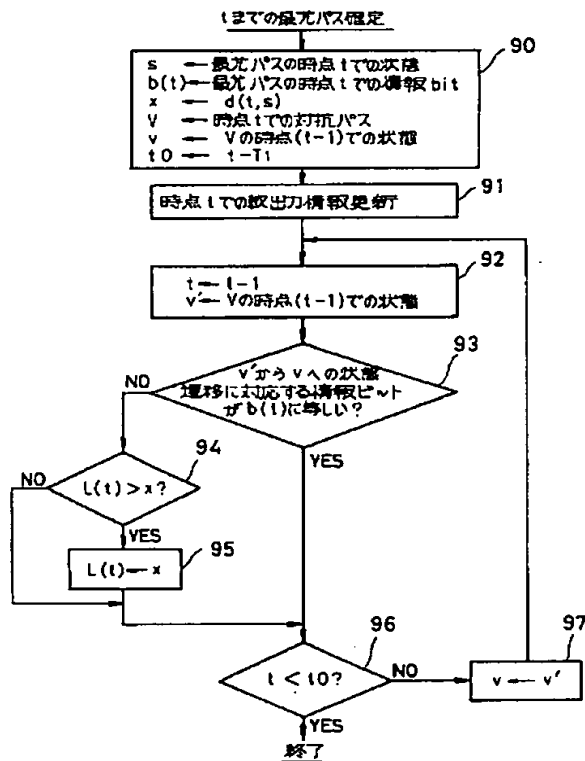
【図 10】



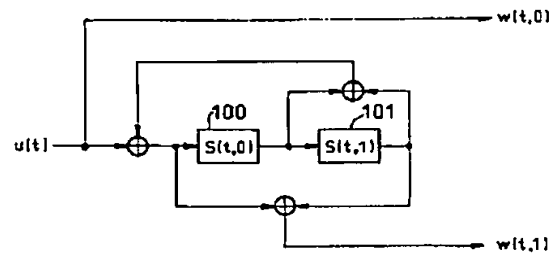
【図 8】



【図 9】



【図 11】



【図 12】

